Attorney's Docket No.: 10559-371001 (P10177)

APPLICATION

FOR

UNITED STATES LETTERS PATENT

TITLE: UPGRADING PERIPHERAL DEVICES

APPLICANT: DERRICK I. HISATAKE

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EL689319359US

I hereby certify that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, D.C. 20231.

1-3-01

Date of Deposit

Signature

Derek W. Norwood

Typed or Printed Name of Person Signing Certificate

UPGRADING PERIPHERAL DEVICES

BACKGROUND

This invention relates to upgrading peripheral devices.

Peripheral devices provide input, output or other auxiliary functions to other devices in a computer system. Often the peripheral devices are embedded within network devices that lack a built-in user interface.

One method of upgrading such peripheral devices includes rebooting the network device into a specialized mode and using a special connector, such as a null modem cable, to connect the network device to a personal computer or other device that has a user interface. Using an application specific to the latter device's operating system, a user can upgrade the peripheral device.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a computer system.

FIG. 2 is a flow chart.

FIG. 3 is a flow chart.

FIG. 4 shows an update package.

FIG. 5 is a flow chart.

DETAILED DESCRIPTION

As shown in FIG. 1, a network device 12 is coupled to a network 16 through which it can communicate with a client device such as a personal computer (PC) 15 connected to the network. The devices 12, 15 communicate by sending and

receiving packages to and from one another. The packages can contain data, executable files, or a combination of both. A file transfer server 14 within the network device 12 helps manage the communications by storing the packages sent by the device 15. A processor 17 executes the functions that appropriately utilize the contents of the packages.

An internal peripheral device 10 is located within the network device 12. Software 11 used by the peripheral device 10 to perform its own functions is stored within the peripheral device. The software 11, which may include firmware, can be stored, for example, as read-only memory (ROM).

As illustrated by FIG. 2, upon receiving 30 a package from the device 15 over the network 16, the file transfer server 14 determines 32 what type of package it is. For example, packages can include a key string, such as a filename extension, that indicates what the package's contents are and the appropriate functions needed to employ them. The server 14 examines the package's identifying key string to determine the package type. If the package is identified as a peripheral device upgrade, the server 14 allocates 34 a memory block 18 and stores 36 the contents of the package in the block. The server 14 then calls 38 a corresponding routine 20 that upgrades the peripheral 10 device with the package.

As illustrated by FIG. 3 the upgrade routine 20 initially causes a communications channel 22 to be opened 40 between the processor 17 and the peripheral device 10. The communications channel can be, for example, a serial connection or a parallel connection. Next, the routine 18 identifies 42 the type and version of peripheral device 10 and its upgrade process using the connection 22. The peripheral device 10 is set 44 to a mode in which its internal software can be upgraded. Setting the peripheral device 10 to the upgrade mode may be accomplished, for example, by sending a command to the device 10, or by other techniques described below.

The routine 18 verifies 46 that the upgrade file in the stored package is the correct type for the peripheral device 10, and then parses the package 48 into a format suitable for transfer to the peripheral device 10. The parsed file is uploaded 50 to the peripheral device 10 over the communications channel 22. After the file is uploaded, a determination is made 52 as to whether the upgrade was completed successfully. If the upgrade was successful, a message reflecting the successful upgrade is sent 54 to the remote device 15. If the upgrade was not successful, for example, if an error fatal to the upgrade procedure occurs during the process, an error message is sent 56 to the remote device 15.

FIG. 4 shows an exemplary upgrade package 60 for upgrading the peripheral device 10. The package 60 contains

software 62 that is intended to replace the peripheral device's 10 old software 11. The package 60 also contains a flash erase file 64 that is used to erase the old software in the peripheral device 10, thus setting the peripheral device 10 to a mode in which the new software 62 can be implemented. The flash erase file 64 can contain all zeros or equivalent null data content used to erase the specified memory location in the peripheral device 10.

FIG. 5 illustrates how the flash erase file 64 can be used to set the peripheral device 10 to the upgrade mode. In this particular implementation, the connection 22 to the peripheral device 10 carries serial data. One line of the flash erase file 64 is read 70 from the memory 18, appended 72 to a packet header and checked 74 to confirm that it is in a valid format. The packet header contains instructions for the peripheral device 10 to overwrite the contents of its software 11 with the contents of the packet. The packet then is sent 76 to the peripheral device 10 over the connection 22. The upgrade routine 20 checks 78 if the flash erase file's 64 transfer is complete, and repeats blocks 70, 72, 74 and 76 until the entire new file is downloaded to the peripheral device 10. If necessary, block 76 can be repeated a predetermined number of times before an error message is sent 56 to the remote device 56.

The automatic upgrade technique described above can be implemented, for example, in network devices such as the Intel® InBusiness™ Internet Station 56K as well as other

devices coupled to a computer network. The process can be used to upgrade embedded peripheral devices such as onboard modems or other devices installed inside the network device. The file server 14 can use, for example, File Transfer Protocol (``FTP'') to send and receive packages over the network 16. Other transfer methods, such as HyperText Transfer Protocol (HTTP), also may be used.

Advantages of these techniques may include the ability for upgrades to be made to embedded peripheral devices without the use of special connections, such as null modem cables, between a client PC and the network device. In addition, the peripheral devices can be upgraded regardless of the operating system on the client device 15. In many implementations the techniques would not require any specialized knowledge of the upgrading process, allowing end users to perform the upgrades without assistance from a service representative. Also, the forgoing techniques can help reduce the amount of time required to upgrade embedded peripheral devices.

Various features of the system can be implemented in hardware, software, or a combination of hardware and software. For example, some aspects of the system can be implemented in computer programs executing on programmable computers. Each program can be implemented in a high level procedural or object-oriented programming language to communicate with a computer system. Furthermore, each such computer program can be stored on a storage medium, such as

read-only-memory (ROM) readable by a general or special purpose programmable computer, for configuring and operating the computer when the storage medium is read by the computer to perform the functions described above.

Other implementations are within the scope of the following claims.